

Ministry of Education and Science of Russian Federation

National Research Tomsk Polytechnic University

Term Project: Premier League Stats Analysis – Manchester United v/s
Manchester City

Students: Cristian R. Machuca

Phana Tavoranpradit

Sheikh Arbid Irfan

Group: 8PM9I

BIG DATA SOLUTIONS

Tomsk – 2019

Table of Content

INTRODUCTION	3
DATASET DESCRIPTION	4
DATA CLEANING	5
PART 1: WINS VS LOSSES (EXPLORATION)	10
A. Extracting data that is valid only for Manchester United and Manchester city out of all the teams in the dataset.	10
B. Slicing the original DataFrame into two.	10
C. Creating the Matplotlib plots	11
PART 2: COMPARATIVE ANALYSIS OF GOALS SCORED	22
CONCLUSION	32
REFERENCE	33

INTRODUCTION

Manchester United is one of the most successful clubs in football history. Sir Alex Ferguson has been the most successful manager for the club. Under his directions between 1986 and 2013 the team won around 38 trophies, after his retirement the team suffered a lack of performance even when renowned coaches like Louis Van Gaal and Jose Mourinho took the technical direction of the team.

On the other hand, during the last years Manchester City has been getting great results. Once referred to as “noisy neighbours”, City have managed to surpass their neighbours in recent years. There has been a lot of investment in both teams but it is looking as if City has made the better investments.

This rivalry is one of the oldest rivalries in football and the objective of this project is to make a comparative analysis of the performance of these two teams from the 2006-2007 season to the 2017-2018 seasons and visualize how the performances change, and discover how their match results have evolved over this period.

We try to find the reasons for City surpassing United in this analysis.

For the creation of this project we used the following tools:

- The Jupyter Notebook to create the code and the visualizations.
- The Python Data Analysis Library (PANDAS) for the exploration and data analysis.
- The NumPy library for the use of mathematical tools and mathematical analysis.
- The Matplotlib library for the generation of charts and visualization of the data.

Dataset Description

Two datasets have been used for this analysis, the first one contains all the results of all the games played in the England Premier League from the 2006/2007 to the 2017/2018 season. It includes the name of the team, wins, draws, losses, goals scored, yellow cards, red cards and touches of each of the 20 participating teams during these seasons.

The second dataset provides information about the results of the matches, and goals scored when the team plays at home or as visitors.

Importing the dataset into Jupyter Notebook:

The next code allows the importation of the dataset for its exploration and analysis:

First, it is important to import all the libraries we will need:

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

Using this code all the missing values if found in the format provided will be recognized :

```
missing_values = ["n/a", "na", "-", ""]
```

```
Data_results = pd.read_csv("../results.csv", na_values=missing_values)
```

```
Data_stats = pd.read_csv("../stats.csv", na_values=missing_values)
```

DATA CLEANING

After importing the data, it is required to verify that the file has been loaded correctly, thus the data can be visualized using the code shown below:

```
data_stats.head()
```

	team	wins	losses	goals	total_yel_card	total_red_card	total_scoring_att	ontarget_scoring_att	hit_woodwork	goals_per_game
0	Manchester United	28.0	5.0	83.0	60.0	1.0	698.0	256.0	21.0	2.97
1	Chelsea	24.0	3.0	64.0	62.0	4.0	636.0	216.0	14.0	2.27
2	Liverpool	20.0	10.0	57.0	44.0	0.0	668.0	214.0	15.0	2.00
3	Arsenal	19.0	8.0	63.0	59.0	3.0	638.0	226.0	19.0	2.10
4	Tottenham Hotspur	17.0	12.0	57.0	48.0	3.0	520.0	184.0	6.0	1.85

5 rows x 42 columns

This gives us the first five entries in the dataset, we can see that our data set stats.csv had 5 rows and 42 columns.

But we also notice this:

touches	big_chance_missed	clearance_off_line	dispossessed	penalty
25686.0	NaN	1.0	NaN	
24010.0	NaN	2.0	NaN	
24150.0	NaN	1.0	NaN	
25592.0	NaN	1.0	NaN	
22200.0	NaN	2.0	NaN	

The data has some NaN values, which could be a problem in future analysis to be performed with this file. So will need to do something about this data. Either just drop the column or fill it up with average values.

The same check for the other dataset called results.csv:

```
data_results.head()
```

	home_team	away_team	home_goals	away_goals	result	season
0	Sheffield United	Liverpool	1.0	1.0	D	2006-2007
1	Arsenal	Aston Villa	1.0	1.0	D	2006-2007
2	Everton	Watford	2.0	1.0	H	2006-2007
3	Newcastle United	Wigan Athletic	2.0	1.0	H	2006-2007
4	Portsmouth	Blackburn Rovers	3.0	0.0	H	2006-2007

This gives an overview of the data that we are going to use, but we need to check for missing values too.

```
print (data_results.isnull().sum())
```

```
home_team    0
away_team    0
home_goals   0
away_goals   0
result        0
season        0
dtype: int64
```

We can see that we have no missing data here.

We can verify this using another command:

```
print (data.isnull().values.any())
```

It returns:

```
False
```

This command will check that our dataset still has missing data or not, if the result is “True” it means we still have missing data but if the result is “False” it means you don’t have missing data left.

So we have no missing data in this dataset.

Now we can focus on the other dataset which had NaN values and missing data and fill it up with average values:

```
print (data_stats.isnull().sum())
```

The result:

goals_conceded	0
saves	20
outfielder_block	0
interception	0
total_tackle	0
last_man_tackle	0
total_clearance	0
head_clearance	20
own_goals	0
penalty_conceded	0
pen_goals_conceded	0
total_pass	0
total_through_ball	20
total_long_balls	0
backward_pass	80
total_cross	0
corner_taken	0
touches	0
big_chance_missed	80
clearance_off_line	0
dispossessed	20
penalty_save	0
total_high_claim	0
punches	0

We will see how much data is missing and from which column.

```
print (data.isnull().values.any())
```

Running this command gives us the following result:

```
True
```

Now we are sure that missing data exists in our dataset so we need to fix this issue:

The total number of missing values is shown by:

```
print (data.isnull().sum().sum())
```

```
In [48]: 1 #total of missing values
          2 print (data.isnull().sum().sum())
          220
```

Now we will fix the missing data issue by filling the columns with missing values with the mean value for this column:

```
data['saves'] = data['saves'].fillna(data['saves'].mean())
```

```
data['head_clearance']
data['head_clearance'].fillna(data['head_clearance'].mean())
```

```
data['total_through_ball']
data['total_through_ball'].fillna(data['total_through_ball'].mean())
```

```
data['backward_pass']
data['backward_pass'].fillna(data['backward_pass'].mean())
```

```
data['dispossessed'] = data['dispossessed'].fillna(data['dispossessed'].mean())
```

After you fixed this issue we can check it by :

```
print (data.isnull().sum().sum())
```



```

: 1 #total of missing values
  2 print (data.isnull().sum().sum())
0

```

All of our missing data is already gone and we can check it by thin command also:

```
print (data.isnull().sum())
```

```

team 0
wins 0
losses 0
goals 0
total_yel_card 0
total_red_card 0
total_scoring_att 0
ontarget_scoring_att 0
hit_woodwork 0
att_hd_goal 0
att_pen_goal 0
att_freekick_goal 0
att_ibox_goal 0
att_obox_goal 0
goal_fastbreak 0
total_offside 0
clean_sheet 0
goals_conceded 0
saves 0
outfielder_block 0
interception 0
total_tackle 0
last_man_tackle 0
total_clearance 0
head_clearance 0
own_goals 0
penalty_conceded 0
pen_goals_conceded 0
total_pass 0
total_through_ball 0
total_long_balls 0
backward_pass 0
total_cross 0
corner_taken 0
touches 0
big_chance_missed 0
clearance_off_line 0
dispossessed 0
penalty_save 0
total_high_claim 0
punches 0
season 0
dtype: int64

```

You will see that all of the columns don't have any missing data anymore.

Now we are ready for future exploratory data analysis of the datasets.

PART 1: Wins Vs Losses (EXPLORATION)

A. Extracting data that is valid only for Manchester United and Manchester city out of all the teams in the dataset.

Once we clean the data, it is necessary to extract the data for analysis purposes and that is the data valid only for Manchester United and Manchester City respectively which is done by the following code:

```
data_united = data[data.team=='Manchester United']
```

```
data_city = data[data.team=='Manchester City']
```

Now all the data for each team are stored into two variables.

B. Slicing the original DataFrame into two.

Data acquisition of the won matches for Manchester United and Manchester City respectively:

Wins v Season.

```
features_wins_united = ['wins','season']
```

```
data_winsVseason_united = data_united[features_wins_united]
```

```
features_wins_city = ['wins','season']
```

```
data_winsVseason_city = data_city[features_wins_city]
```

And another for Losses v Season.

Data acquisition of the lost matches:

```
features_loss_united= ['losses','season']
```

```
data_lossVseason_united = data_united[features_loss_united]
```

```
features_loss_city= ['losses','season']
```

```
data_lossVseason_city = data_city[features_loss_city]
```

C. Creating the Matplotlib plots

The data is ready visualization and, the preparation for visualizing the data, we will use the matplotlib library for visualizing the data for number of wins and losses for Manchester United under Sir Alex Ferguson and then subsequent successors to showcase the change in form:

```
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [12, 8]

plt.plot(data_winsVseason_united.season,      data_winsVseason_united.wins,
label = 'Wins', marker='o')

plt.plot(data_lossVseason_united.season,      data_lossVseason_united.losses,
label = 'Losses', marker='o')

plt.ylim(0,30)

plt.grid(which='major', axis='both', linestyle='-.', linewidth=0.75)

plt.xticks(rotation=60, fontsize=14)

plt.yticks(fontsize=14)

plt.legend(fontsize=14)

plt.title('Wins and Losses v/s Season for Manchester United', fontsize = 20)

try:

    plt.annotate('Moyes comes in', xy=('2013-2014',19), xytext=('2013-2014',
21),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
    fontsize =10)

    plt.annotate('Moyes comes in', xy=('2013-2014',12), xytext=('2013-2014',
14),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
    fontsize =10)
```

```
plt.annotate('Van Gaal comes in', xy=('2014-2015',20), xytext=('2014-2015', 23),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8), fontsize =10)
```

```
plt.annotate('Van Gaal comes in', xy=('2014-2015',8), xytext=('2014-2015', 11),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8), fontsize =10)
```

```
plt.annotate('Mourinho comes in', xy=('2016-2017',18), xytext=('2016-2017', 20),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8), fontsize =10)
```

```
plt.annotate('Mourinho comes in', xy=('2016-2017',5), xytext=('2016-2017', 8),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8), fontsize =10)
```

```
plt.text('2009-2010',16,'The Famous SAF Era', horizontalalignment='center', verticalalignment='center', fontsize =16, bbox=dict(facecolor='red', alpha=0.5))
```

except TypeError:

```
pass
```

This code gives us the following graphical representation:

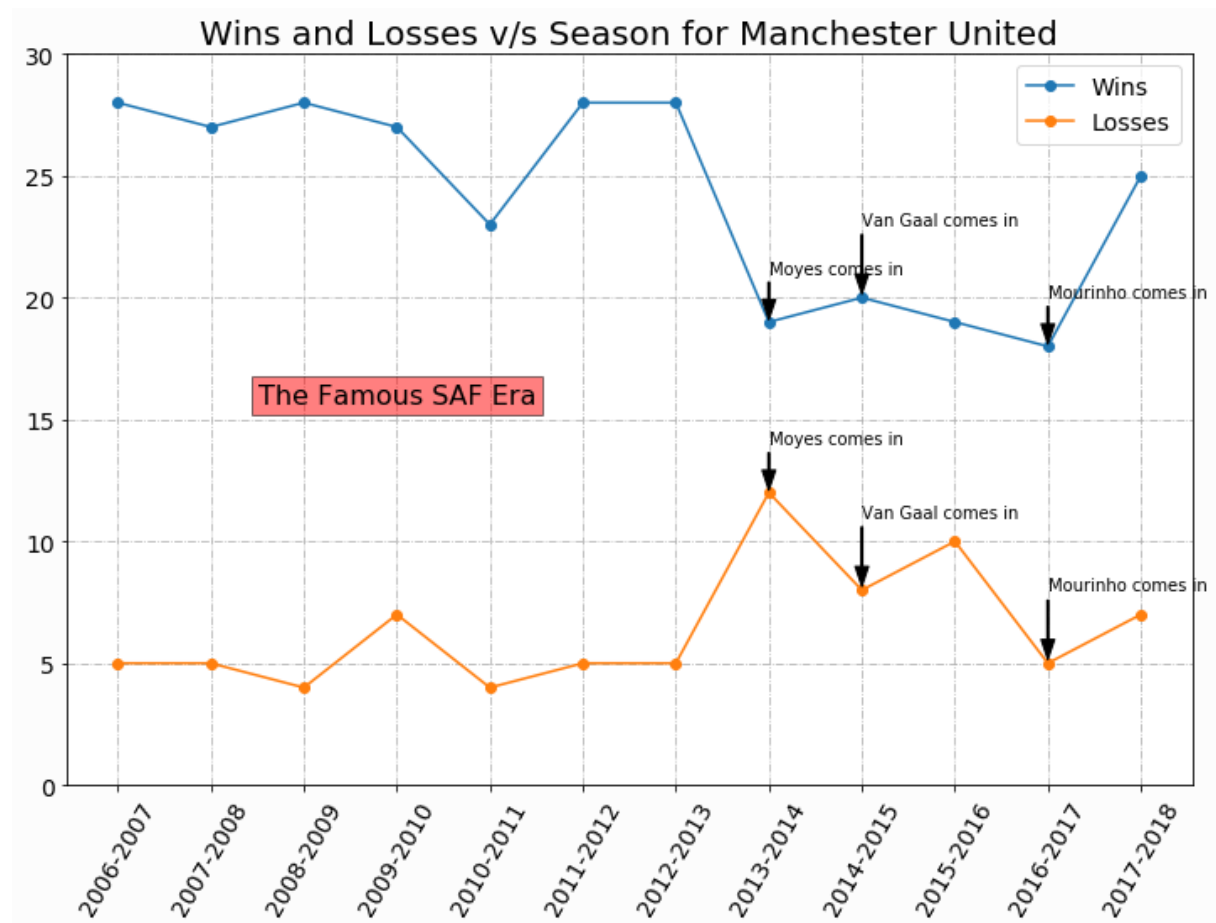


Figure 1 Manchester United

From the chart above it is quite clear that after the departure of Sir Alex Ferguson, United have suffered by a very low amount of wins each season. We can see the consistency in the amount of wins under Sir Alex and sudden loss after his departure. Interestingly, the United team changed a number of coaches after Sir Alex, who was in charge from 1986 to 2013 (21 seasons!) United have changed coaches quite frequently, almost year after year. But none of them have been able to get them to the same height and consistency as Sir Alex Ferguson.

Now, we use the same way to visualize the number of wins and losses over the seasons for Manchester City and compare the two plots:

Code:

```
plt.rcParams['figure.figsize'] = [12, 8] #creates a graph of 10x7 inches
plt.plot(data_winsVseason_city.season, data_winsVseason_city.wins, label =
'Wins', marker='o') #plot Wins v Season
plt.plot(data_lossVseason_city.season, data_lossVseason_city.losses, label =
'Losses', marker='o') #plot Loss vs Season, with markers and label
plt.ylim(0,33) #for setting y limits from 15 to 30.
plt.grid(which='major', axis='both', linestyle='-.', linewidth=0.75) #plotting grid.
See pyplot.grid in Google.
plt.xticks(rotation=60, fontsize=14) #rotate x axis labels and increase font.
plt.yticks(fontsize=14) #increase yticks fontsize.
plt.legend(fontsize=14) #show labels as legend
plt.title('Wins and Losses v/s Season for Machester City', fontsize = 20)
```

try: #try catch is used because here x-coord is a string like 2006-2007 and hence gives a TypeError. So TypeError errors are overlooked in except statement.

```
plt.annotate('Sheikh buys club', xy=('2007-2008',15), xytext=('2007-2008',
21),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

#For the point when Arab investment came in

#xy is the position of marker point. xytext is position of label text.

```
plt.annotate('FA Cup final', xy=('2010-2011',21), xytext=('2010-2011',
23),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

#For winning FA Cup final

```
plt.annotate('£100 million spent', xy=('2009-2010',18), xytext=('2009-2010',
20),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

#For when city spent 100 million

```
plt.annotate('Premier League title', xy=('2011-2012',28), xytext=('2009-2010',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
#For PL title
```

```
plt.annotate('Premier League title', xy=('2013-2014',27), xytext=('2013-2014',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
#For PL title
```

```
plt.annotate('Premier League title', xy=('2013-2014',27), xytext=('2013-2014',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
#For PL title
```

```
plt.annotate('Pep Guardiola comes in', xy=('2015-2016',19), xytext=('2015-
2016', 22 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1,
headwidth=8), fontsize =10)
#For winning curve when Guardiola comes in.
```

```
plt.text('2009-2010',16,'The Overhaul of City', horizontalalignment='center',
verticalalignment='center', fontsize =16, bbox=dict(facecolor='red', alpha=0.5))
#For the middle big text
```

```
plt.annotate('Highest Points finish', xy=('2017-2018',32), xytext=('2017-2018',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
#For highest point finish under Guandiola
```

```
except TypeError:
pass
```

This code gives us the following representation of data for Manchester City:

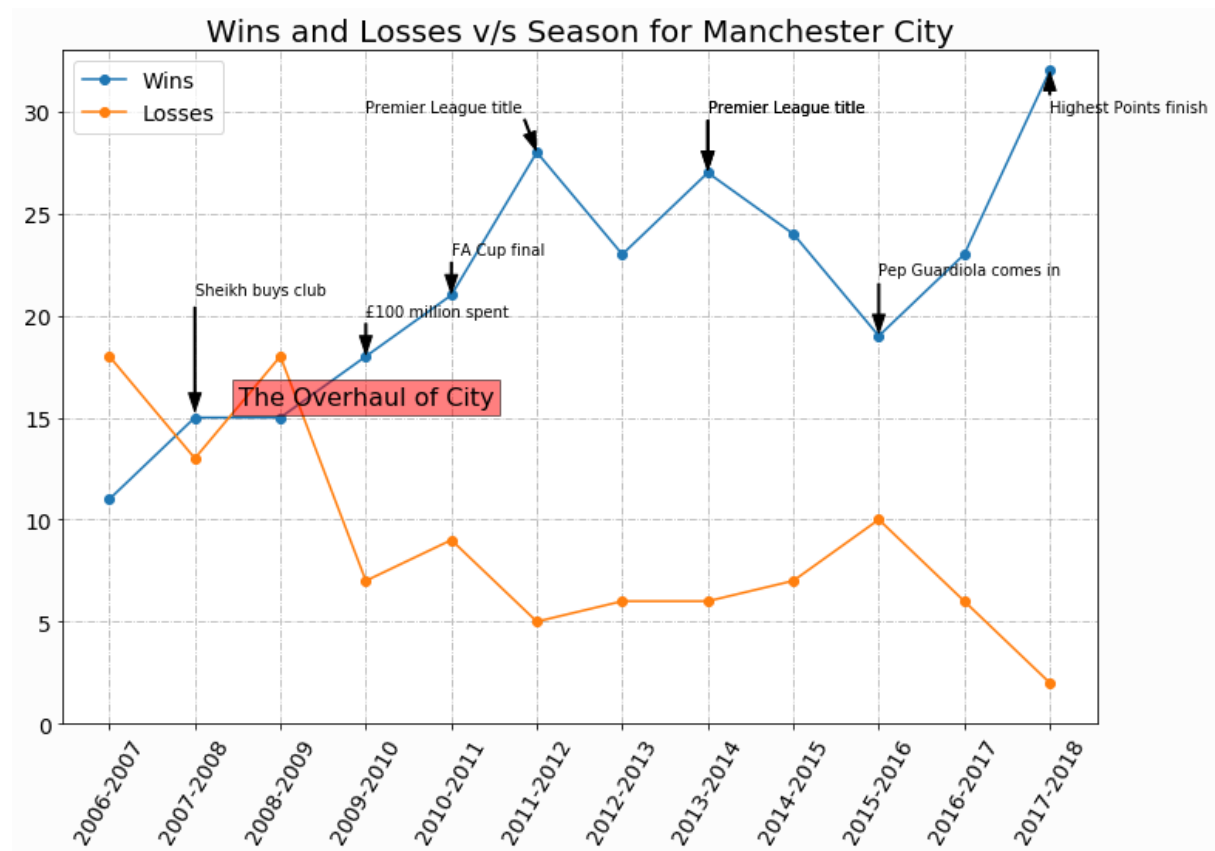


Figure 2 Manchester City

From the chart we can gather the information that, after the purchase of the club in 2008, by an Arab investor and subsequent transfer expenditure over the years, City have been doing much better than before. Even winning many trophies and PL titles over the years.

Now let's merge these two charts together, only for wins per season so we can compare it for each season:

Code:

```
import matplotlib.pyplot as plt
```

```
#Plotting Wins and Losses v/s Season for Manchester United
```

```
plt.rcParams['figure.figsize'] = [14, 10] #creates a graph of 10x7 inches
```



```
plt.plot(data_winsVseason_united.season, data_winsVseason_united.wins,
label = 'Wins United', marker='o') #plot Wins v Season
#plt.plot(data_lossVseason_united.season, data_lossVseason_united.losses,
label = 'Losses United', marker='o') #plot Loss vs Season, with markers and label
plt.ylim(0,30) #for setting y limits from 15 to 30.
plt.grid(which='major', axis='both', linestyle='-.', linewidth=0.75) #plotting grid.
See pyplot.grid in Google.
plt.xticks(rotation=60, fontsize=14) #rotate x axis labels and increase font.
plt.yticks(fontsize=14) #increase yticks fontsize.
plt.legend(fontsize=14) #show labels as legend
```

```
plt.rcParams['figure.figsize'] = [12, 8] #creates a graph of 10x7 inches
plt.plot(data_winsVseason_city.season, data_winsVseason_city.wins, label =
'Wins City', marker='o') #plot Wins v Season
#plt.plot(data_lossVseason_city.season, data_lossVseason_city.losses, label =
'Losses City', marker='o') #plot Loss vs Season, with markers and label
plt.ylim(0,33) #for setting y limits from 15 to 30.
plt.grid(which='major', axis='both', linestyle='-.', linewidth=0.75) #plotting grid.
See pyplot.grid in Google.
plt.xticks(rotation=60, fontsize=14) #rotate x axis labels and increase font.
plt.yticks(fontsize=14) #increase yticks fontsize.
plt.legend(fontsize=14) #show labels as legend
plt.title('Wins per seasons Season for City v/s United', fontsize = 20)
```

try: #try catch is used because here x-coord is a string like 2006-2007 and hence gives a TypeError. So TypeError errors are overlooked in except statement.

```
plt.annotate('Moyes comes in', xy=('2013-2014',19), xytext=('2013-2014',
21),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

#For the wins curve. The label is Moyes comes in.

#xy is the position of marker point. xytext is position of label text.

```
plt.annotate('Van Gaal comes in', xy=('2014-2015',20), xytext=('2014-2015',
23),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For wins curve when LVG comes in.
```

```
plt.annotate('Mourinho comes in', xy=('2016-2017',18), xytext=('2016-2017',
20),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For wins curve when Mou comes in.
```

```
plt.annotate('Sheikh buys club', xy=('2007-2008',15), xytext=('2007-2008',
21),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For the point when Arab investment came in
```

```
#xy is the position of marker point. xytext is position of label text.
```

```
plt.annotate('FA Cup final', xy=('2010-2011',21), xytext=('2010-2011',
23),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For winning FA Cup final
```

```
plt.annotate('£100 million spent', xy=('2009-2010',18), xytext=('2009-2010',
20),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For when city spent 100 million
```

```
plt.annotate('Premier League title', xy=('2011-2012',28), xytext=('2009-2010',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For PL title
```

```
plt.annotate('Premier League title', xy=('2013-2014',27), xytext=('2013-2014',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For PL title
```

```
plt.annotate('Premier League title', xy=('2013-2014',27), xytext=('2013-2014',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For PL title
```

```
plt.annotate('Pep Guardiola comes in', xy=('2015-2016',19), xytext=('2015-
2016', 22 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1,
headwidth=8), fontsize =10)
```

```
#For winning curve when Guardiola comes in.
```

```
plt.annotate('Highest Points finish', xy=('2017-2018',32), xytext=('2017-2018',
30 ),arrowprops=dict(facecolor='black', shrink=0.05, width=1, headwidth=8),
fontsize =10)
```

```
#For highest point finish under Guardiola
```

```
except TypeError:
```

```
Pass
```

The Code gives us the following representation which will allow us to do a more detailed comparison:

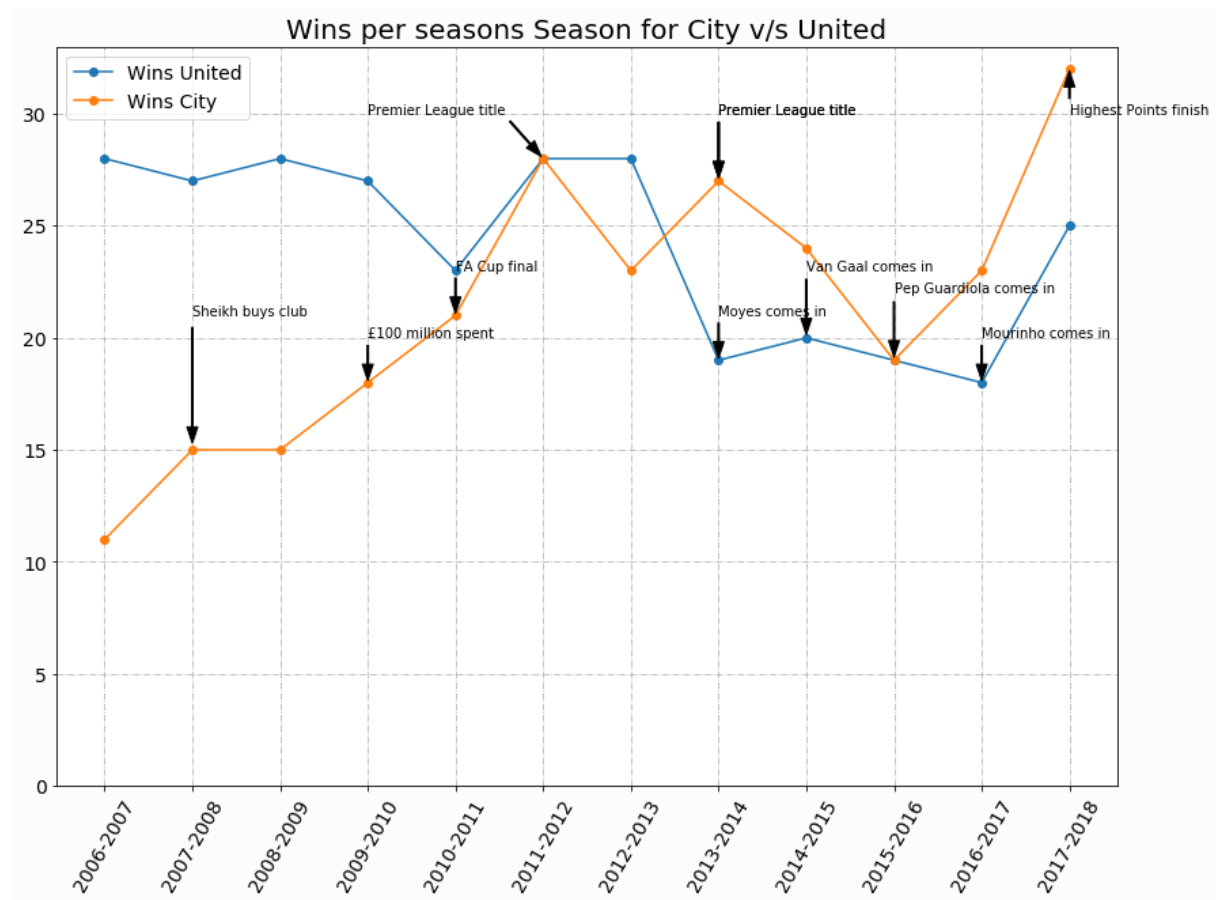


Figure 3 Manchester United v/s Manchester City

According to the data presented above, in terms of wins each season since the 2006-2007 season of the Premier League, it is clear that after the departure of Sir Alex Ferguson in 2013 and his replacement by David Moyes, there has been a significant decrease in form for Manchester United which is highlighted by the sharp decrease in the number of wins while as after 2008 when The Abu Dhabi United Group become the new owners of Manchester City, and the record transfer spending of 100 million Euros, for Manchester City it is quite clear that around the same time interval, there is a significant increase in form highlighted by the amount of Premier League titles they have won and the massive amount of wins in the 2017-2018 season.

We can see how the graph begin by showing the amount of wins for City which are very low and how City gradually catch up with United and finally overshadow them.

Could the sole reason for this change is the huge amount of money spent on player transfers by the club because of Arab investment?

We wanted to find out but couldn't find a free dataset that could allow us to visualize the data for transfer money spent by the two clubs and try to compare, so we used a website to gather relevant data for our analysis. We found that:




#	Club	Total market value ↑	Purchase value ↓	Difference ↑
1	 Man City	€1.30bn	€956.15m	€340.35m
2	 Man Utd	€748.25m	€688.93m	€59.32m
3	 Liverpool	€1.18bn	€544.75m	€634.50m
4	 Arsenal	€677.75m	€454.90m	€222.85m
5	 Chelsea	€842.75m	€400.90m	€441.85m
6	 Everton	€495.70m	€394.25m	€101.45m

Figure 4 Transfer spending by both clubs

Manchester United and Manchester City have spent the most money in the Premier League transfer market. While City are first, United are not far behind at almost 700 million Euros.

So, we cannot say that transfer market spending is the reason for City doing so well in the premier league.

More analysis is needed.

In 2010 Manchester City enjoy their best ever Premier League campaign in their history, finishing 5th and qualifying for the Europa League in the process, as well as reaching the League Cup semi-finals. In 2011, they finally lift the FA Cup, a first trophy is a long time in their history, and they qualify for the 2011/12 Champions League for the first time too. In 2012 City win the League Championship for the first time in 44 years following a thrilling Premier League campaign. In 2014, the same year David Moyes replaces Manchester United City clinch two trophies in one season for the first time since 1970, lifting the Capital

One Cup and Premier League title while as Manchester United suffer a dump in form which they haven't recovered from since.

PART 2: Comparative Analysis of goals scored

The first step is converting the season to a single year due to this way the data is easier to handle and as we can see in the picture shown below the home goals and away goals have float format, thus they will be changed to int format.

	home_team	away_team	home_goals	away_goals	result	season
0	Sheffield United	Liverpool	1.0	1.0	D	2006-2007
1	Arsenal	Aston Villa	1.0	1.0	D	2006-2007
2	Everton	Watford	2.0	1.0	H	2006-2007
3	Newcastle United	Wigan Athletic	2.0	1.0	H	2006-2007
4	Portsmouth	Blackburn Rovers	3.0	0.0	H	2006-2007

The next snippet of code allows to convert the season.

```
arr = list(data_results['season'])
```

```
season = list()
```

```
for val in arr:
```

```
    temp = val.split('-')[0]
```

```
    season.append(temp)
```

```
season
```

```
season = pd.to_datetime(season)
```

```
season.year
```

```
data_results['season'] = season.year
```

```
data_results['season'] = data_results['season'].astype(int)
```

```
data_results.head(5)
```

Once applied these lines of code these are the results:

	home_team	away_team	home_goals	away_goals	result	season
0	Sheffield United	Liverpool	1.0	1.0	D	2006
1	Arsenal	Aston Villa	1.0	1.0	D	2006
2	Everton	Watford	2.0	1.0	H	2006
3	Newcastle United	Wigan Athletic	2.0	1.0	H	2006
4	Portsmouth	Blackburn Rovers	3.0	0.0	H	2006

The next snippet of code allows to change the format for the home goals and away goals columns.

```
data_results['home_goals'] = data_results['home_goals'].astype(int)
```

```
data_results['away_goals'] = data_results['away_goals'].astype(int)
```

And now this is our data:

	home_team	away_team	home_goals	away_goals	result	season
0	Sheffield United	Liverpool	1	1	D	2006
1	Arsenal	Aston Villa	1	1	D	2006
2	Everton	Watford	2	1	H	2006
3	Newcastle United	Wigan Athletic	2	1	H	2006
4	Portsmouth	Blackburn Rovers	3	0	H	2006

Now the data is ready for being analyzed, the next function allows to know the goals scored as home and visitor team for Manchester United and Manchester City.

```
def goals(season):
```

```
    a=data_results[(data_results.season==season)&(data_results.home_team=='Manchester United')]['home_goals'].sum()
```

```
    b=data_results[(data_results.season==season)&(data_results.away_team=='Manchester United')]['away_goals'].sum()
```

```
    c=data_results[(data_results.season==season)&(data_results.home_team=='Manchester City')]['home_goals'].sum()
```

```
    d=data_results[(data_results.season==season)&(data_results.away_team=='Manchester City')]['away_goals'].sum()
```

```
    index=['MANCHESTER UNITED','MANCHESTER CITY']
```

```
    m1=[a,c]
```

```
    m2=[b,d]
```

```
    print ("Goals scored season:",season)
```

```
    df = pd.DataFrame({'Home Goals':m1,'Away Goals':m2}, index=index)
```

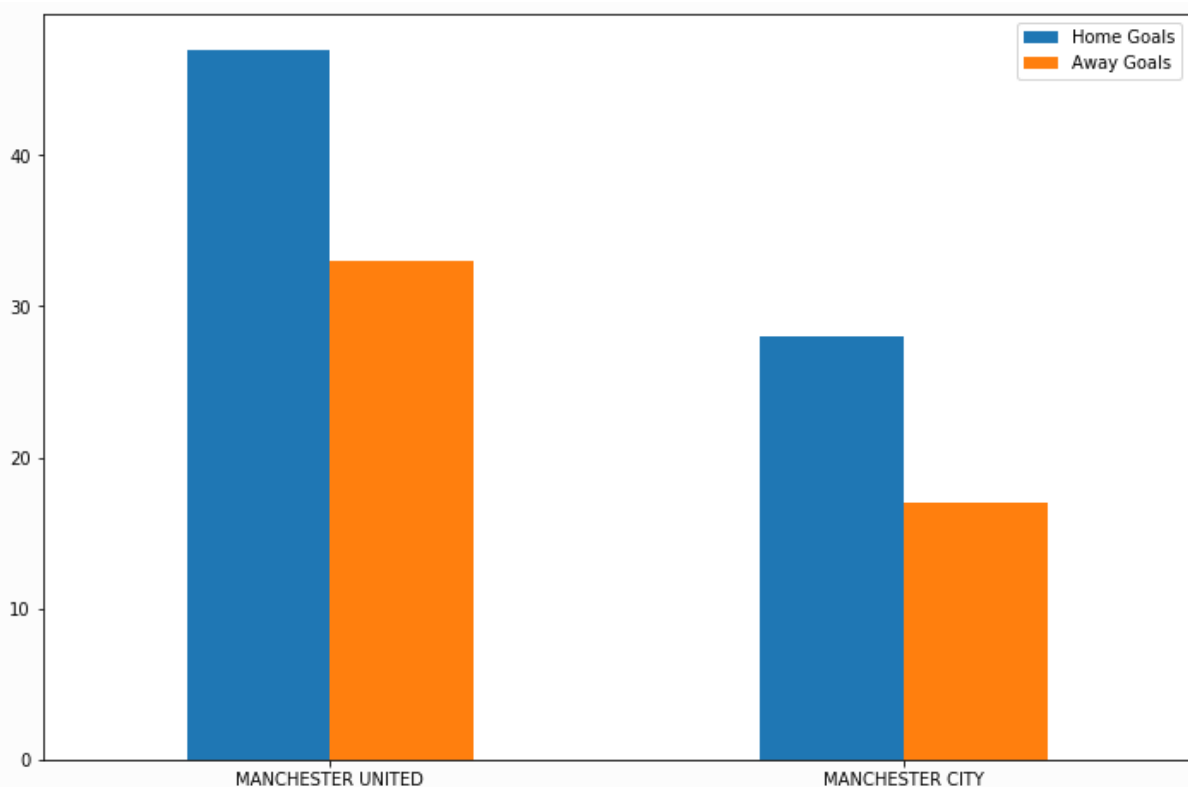


```
ax = df.plot.bar(rot=0)
```

It seems obvious that there is a link between number of goals scored and success in football, that is why we will take some seasons to analyze how the number of goals scored affect the position of the team in the standings at the end of the season.

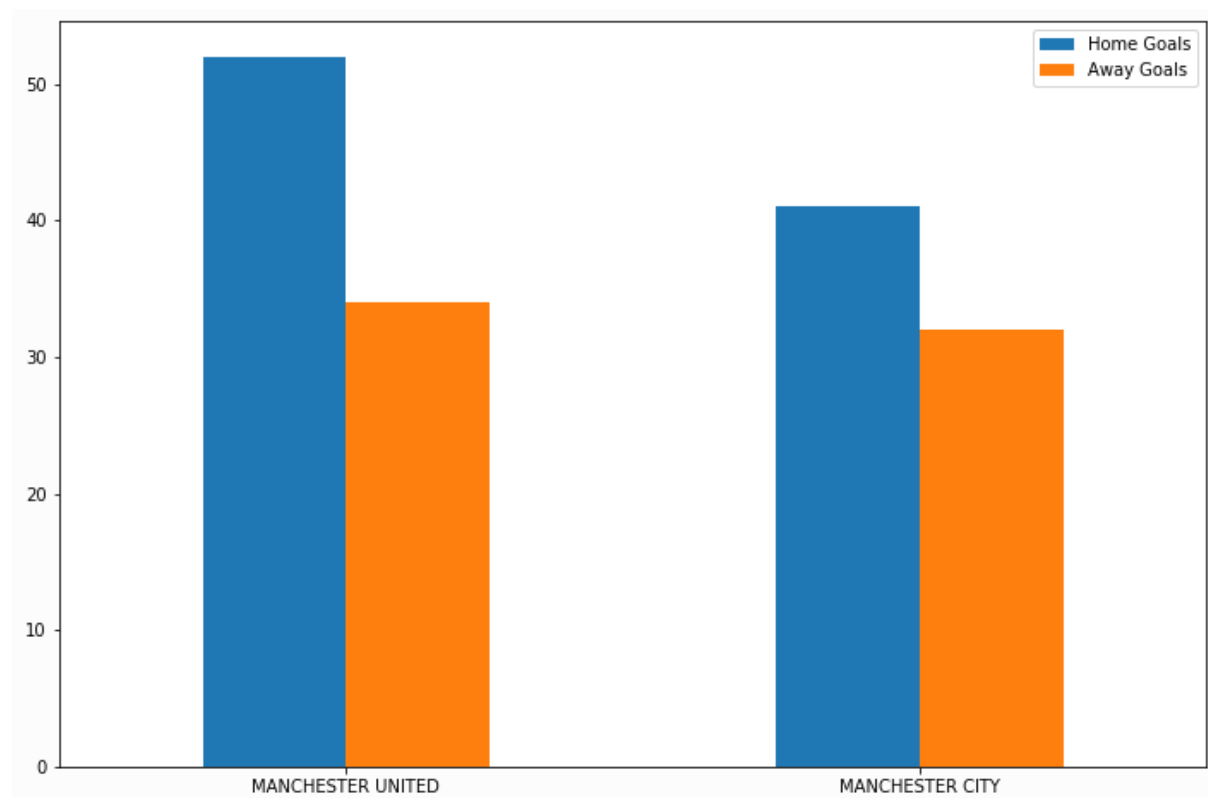
Season 2007, the beginning:

This season Manchester United was the champion and Manchester city ended in the position number 9, we can notice clearly that the more goals scored the better position in the standings.



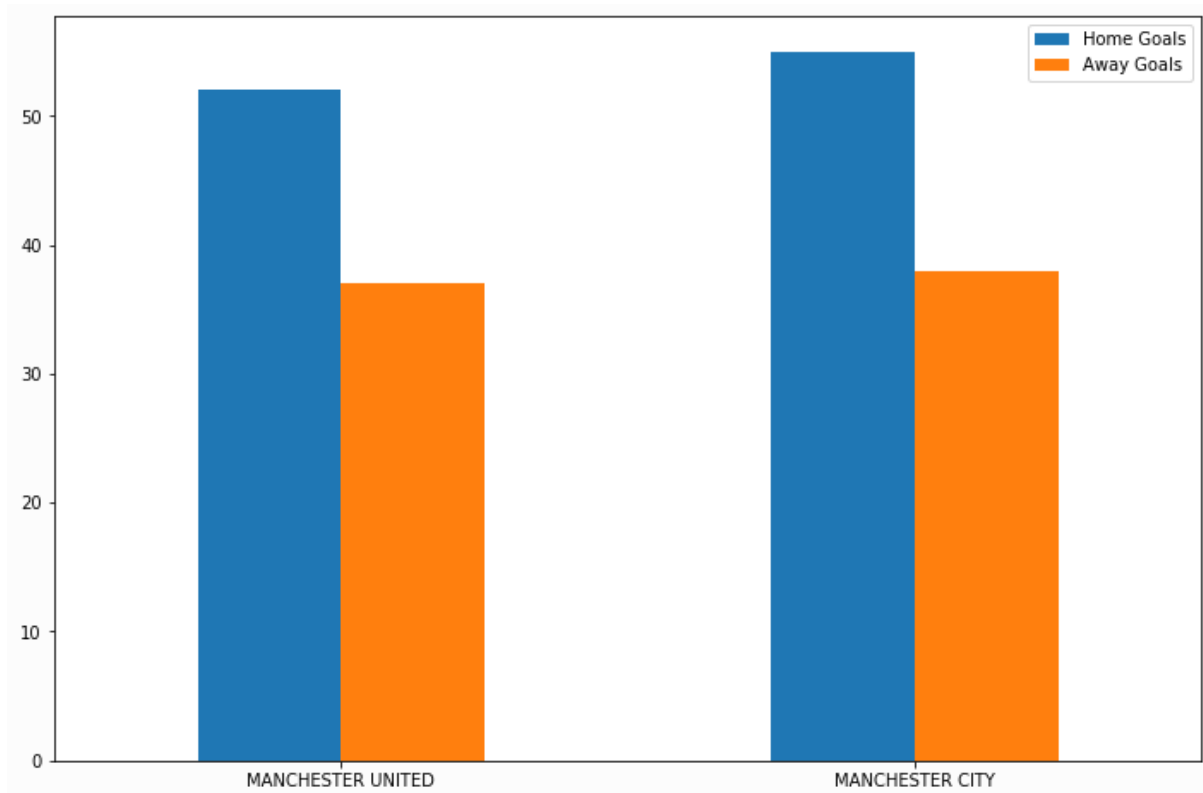
Season 2009, a new team was born:

The owners of the team have decided to turn it into a star squad, 100 million euros are invested in new hires and the results begin to be noticed, Manchester City improves its performance compared to past seasons, although it is still not enough to compete against Manchester United than under Sir Alex Ferguson's command remains a fearsome team with a high number of goals scored per season. Manchester United occupied the second position whereas Manchester City occupied the fifth place.



Season 2011, the break point:

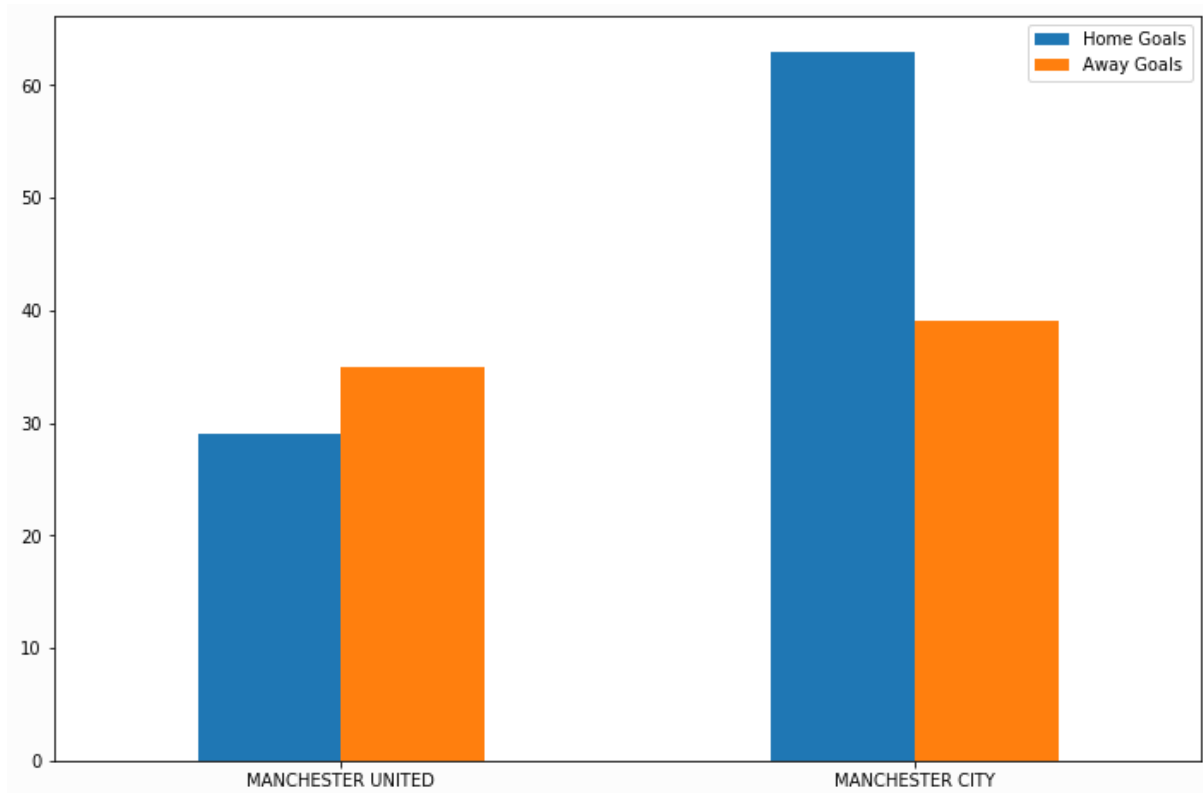
There are no performance differences between the teams of the city of Manchester, the citizen team finishes first and the red evils finish second, only 4 goals scored make Manchester City the most scoring team of the season, despite having finished with the same amount of points as Manchester United.



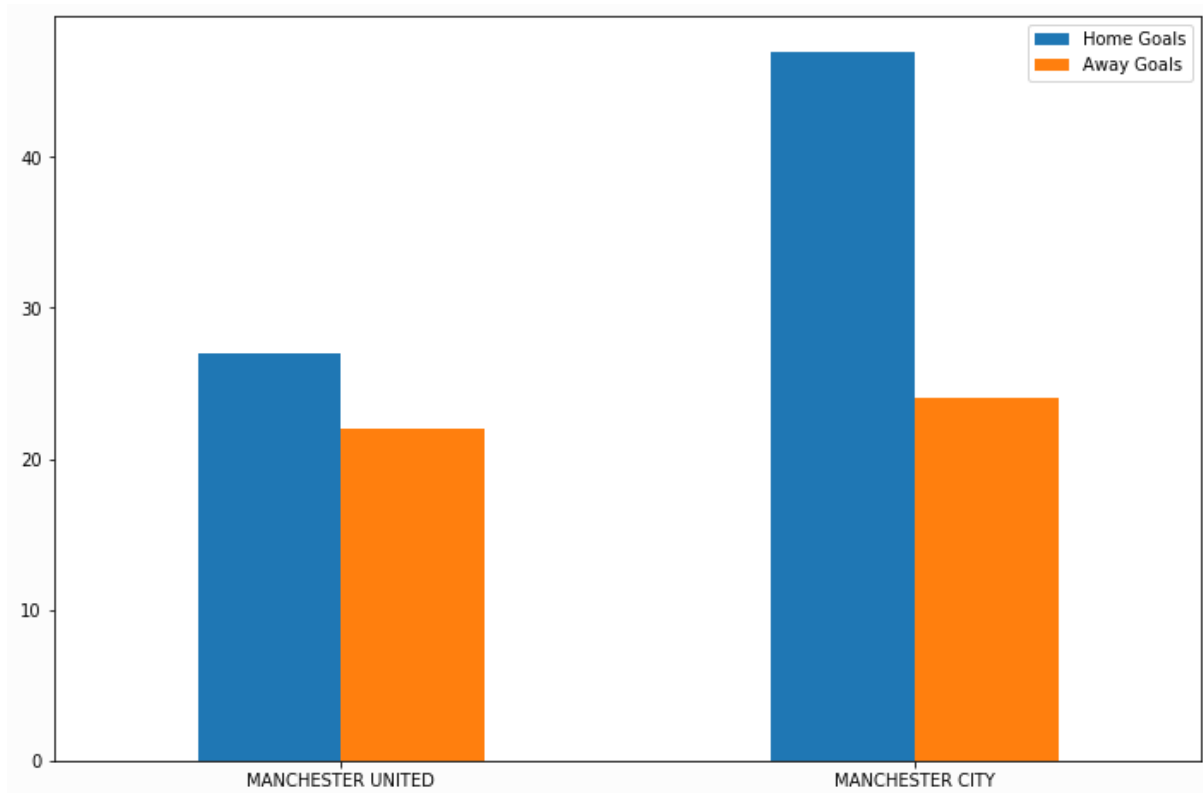
Season 2013, the rising of a great team and the fall of another.

Sir Alex Ferguson has given up the technical direction after almost three decades of success, and the team feels his departure, this is his worst performance in years, and its neighbor and traditional rival seems to have no limit in terms of his performance, the difference has been reversed and now Manchester City makes a big difference with respect to its rivals obtaining the title of the league for that season.

Manchester United finishes the season in the seventh place without the possibility of playing continental tournaments, the goals scored have been insufficient to obtain the expected results.

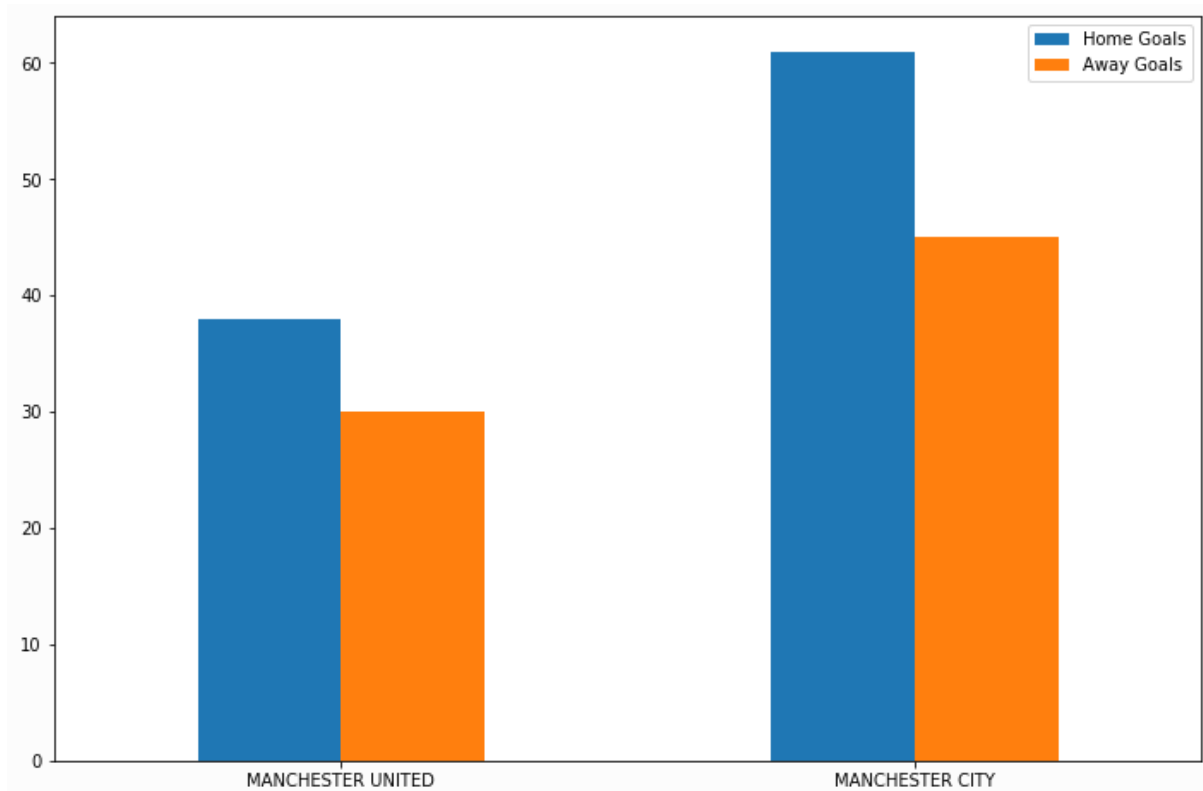


Season 2015, the Manchester City occupied the fourth position and Manchester United the fifth place in the general board of positions, the dominance of the blue team is undeniable, but despite having been the team with the highest number of goals scored during the season it was not able to crown the tournament champion.



Season 2017, a new era.

Manchester City is the total dominator of the premier league, Manchester United simply cannot compete against such high performance of its rival, a new era in the league has begun to be written.



Overall analysis about goals Scored.

As you can be seen in the previous graphics, during the last seasons Manchester United has had a significant decrease in its performance in terms of goals scored, while Manchester city has had a superlative performance, even breaking record goals. Scoring more goals means having a better chance of winning games, but this does not imply that the team with the highest scores will end up being the champion at the end of the season. During the seasons that the blue team has scored more goals than the red devils have finished in a better position in the standings, however some teams with lower goals scored have ended above them.

The goals scored play a fundamental role in determining the performance of the teams, but when it is necessary to determine the overall performance, that means, wins, losses, points obtained and final position when the season ends,

various indicators must be taken into account as goals received , distribution of goals, number of goal chances generated, etc.

Conclusion

In the first part of the analysis, we found that Manchester United have been slumping since the departure of their long-serving manager, Sir Alex Ferguson. And at the same time the fierce rivals Manchester City have been on an up and performing much better and winning important titles. We tried to find the reasons for this big gap in terms of this rivalry between the two teams. We surveyed for the reasons and found that Manchester United fans associate this gap to the purchase of Manchester City by an Arab investment group but we also found that the two teams have spent almost an equal amount of money on acquisition of players. So, the huge influx of money in Manchester City cannot be the sole reason for City doing much better. So we needed more analysis.

In the second part of our analysis, we found that since the 2009 season the performance in terms of goals scored by Manchester City has exceeded the highest expectations, which implies that the team has had good results winning some championships during this time, however the goals scored by Manchester United has fallen during the last seasons, which implies that it has not obtained the desired results and therefore has not won as many trophies as has Manchester City and could not shine as in previous years when Sir Alex Ferguson was in command of the team.

Therefore, we can conclude from our analysis that the sole reason for the big difference in the performance of the two teams is better recruitment of managers by Manchester City which allowed them to build a better team to be able to score a significantly more amount of goals than Manchester United, thereby disproving the popular theory that Manchester City is successful only because of the supposed “oil-money” from Saudi Arabia. So, we can conclude that Manchester United just haven’t found a better successor to Sir Alex Ferguson.

REFERENCE

- Dataset: <https://www.kaggle.com/zaemnalla/premier-league>
- Wikipedia :
 - https://en.wikipedia.org/wiki/Manchester_derby
 - https://en.wikipedia.org/wiki/Manchester_United_F.C.
 - https://en.wikipedia.org/wiki/Manchester_United_F.C.